

UNITED STATES PATENT APPLICATION

FOR

METHOD AND SYSTEM FOR WORKLOAD BALANCING IN A
NETWORK OF COMPUTER SYSTEMS

Inventors:

James Pickel
Shivram Ganduri

Sawyer Law Group LLP
2465 E. Bayshore Road
Suite 406
Palo Alto, CA 94303

2015-05-10 10:00:00

METHOD AND SYSTEM FOR WORKLOAD BALANCING IN A NETWORK OF COMPUTER SYSTEMS

FIELD OF THE INVENTION

5 The present invention relates generally to a remote database managed by a set of database management system servers, and more particularly, to a method and system for enabling a client system to gain access to the remote database for workload balancing and resynchronization.

BACKGROUND OF THE INVENTION

10 In today's information technology industry, customers desire systems that have better performance and cost improvements over their current systems, while permitting continued use of their existing applications. High availability is also important given today's rising mobile workforce and ability for global connectivity through the Internet. Current processing system designs for information technology applications must be scalable, and provide granular growth and continuous application availability. Clustering technology, such as IBM Corporation's S/390 Parallel Sysplex, has been developed to achieve these design needs.

15 A cluster links individual data processing systems in a manner that creates a single, powerful system, which can support numerous users simultaneously. A cluster offers high availability and excellent scalability. If one system/node in the cluster fails, the other systems in the cluster continue to run.

20 Cluster technology provides many advantages in a parallel database management system. For example, Figure 1 illustrates a sysplex environment 100 of three computers

101-103 sharing disk space such as a pool 110 of storage devices, e.g., disk drives, 111-113 where a database resides. Each computer 101-103 includes a database management system (DBMS) member 121-123 and an associated log dataset 151-153 for managing a commit or roll back of a unit of work. Each log dataset 151-153 can only be accessed by the DBMS member that owns it. Each DBMS member 121-123 knows how to communicate with other DBMS members, and each knows how to manage the pool of data 110 that is common to them. A coupling facility 120 links the computers 101-103 to one another and to the pool 110 of storage devices 111-113. The coupling facility 120 also includes a Work Load Manager (WLM) 124 that manages, monitors and balances the workload of each DBMS member 121-123.

The sysplex 100 is accessible to a plurality of client systems 140 via a network connection 145, typically a TCP/IP network. For clarity, only one client 140 is illustrated in Figure 2. The client 140 could be another parallel sysplex or a workstation or other personal computer. The client 140 views the sysplex 100 as one image, i.e., the client 140 views the sysplex as a single remote database, and not as a plurality of discrete DBMS members 121-123. When the client 140 wishes to submit a request or command to the database, the client 140 need only initiate communication with one DBMS member, e.g. 121, of the sysplex 100. In order to gain access to the one member 121, the client 140 sends a message to a domain name server (DNS) 130 asking for a network address, e.g. IP address, of a computer 101-103 in the sysplex 100 containing an active DBMS member 121-123.

The DNS 130 is tightly coupled to, i.e., integrated with, the Work Load Manager (WLM) 124, and keeps a directory of all of the different groups of computers, i.e. sysplexes, on the network, and their respective IP addresses. Each group of computers is given a

domain name, such as "system1.vnet.ibm.com," and each computer *in the group* has a separate IP address. The client 140 registers with the DNS 130 that its databases are part of the system called "system1.vnet.ibm.com." Because the DNS 130 maintains the IP addresses, the client 140 is not required to hard code the IP addresses of the individual computers. Thus, when the client 140 submits its request to connect to a database to the DNS 130, the client 140 identifies the domain name of the database location and the DNS 130 resolves the domain name to a given group of IP addresses.

In further response to the client's 140 request, the DNS 130 checks with the WLM 124 to determine which DBMS members 121-123 are active and on which computers 101-103 they reside. The DNS 130 will then send a message back to the client 140 containing the IP address of a computer 101, 102, 103 that contains an active DBMS member for that named group. In a sysplex environment supporting workload balancing, the IP address returned by the DNS 130 to the client 140 is that of the computer having the least loaded DBMS member. In order to determine the least loaded member, the DNS 130 communicates with the WLM 124, which also monitors the workload of each DBMS member 121-123. The client 140 then utilizes that IP address to establish an initial connection with the corresponding computer containing the DBMS member in the sysplex.

Once the initial connection is made between the client 140 and the one DBMS member, e.g., 121, the DBMS member 121 returns to the client 140 a list (not shown) of active DBMS members and their respective workloads, i.e., a weighted list. The list allows the client to distribute work evenly across the active members of the database sysplex.

This configuration provides data sharing and workload balancing for a plurality of clients. However, the tight coupling of the DNS 130 to the WLM 124 presents several

problems. First, DNSs have not been installed on many sysplex systems and customers are reluctant to integrate a DNS into their existing TCP/IP networks. Without the DNS 130, the client 140 cannot establish contact with the database and the distributed sysplex strategy fails. Second, even if a WLM enabled DNS is installed, a latency is associated with the exchange of information between the WLM and the DNS. Thus, if the client 140 is a gateway sending multiple requests to the DNS 130, the delay due to the latency can impair performance. Third, because the DNS is typically a UNIX based application and the WLM is a database application, integration between the WLM and the DNS is complicated and difficult.

Accordingly, a need exists for a method and system that supports workload balancing in a sysplex system and improves the performance of such a system. The method and system should be easily implemented and should not require extensive modifications of existing systems. The present invention addresses such a need.

SUMMARY OF THE INVENTION

The present invention is directed to a method and system for connecting a client to a database managed by a network of computer systems having a plurality of database management system (DBMS) members. The method and system includes providing a shared network address for the plurality of DBMS members, which is used by the client to connect to an active DBMS member of the plurality of DBMS members.

Through the aspects of the present invention, the shared network address allows the client to connect, via a network router, to any one of the plurality of DBMS members so long as one member is active. Once the client is connected to one member, the member sends to the

client a list of all active members and their respective workloads. The list also includes member-specific network addresses corresponding to each member so that the client can distribute work evenly across the active members.

Through another aspect of the present invention, the member-specific network address is unique to the corresponding DBMS member and is linked with that member, and not to the computer on which it may reside. Thus, if a computer on which the DBMS member is residing goes down or if the DBMS member goes down, and the member is restarted on a different computer, the client can perform a resynchronization process, e.g., commit or roll back, by utilizing the member-specific network address to reconnect to the appropriate member.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a sysplex environment known in the art.

Figure 2 illustrates a sysplex environment in accordance with a preferred embodiment of the present invention.

Figure 3 is a flow chart illustrating the process of work load balancing in accordance with the present invention.

DETAILED DESCRIPTION

The present invention relates generally to a remote database managed by a set of database management system servers, and more particularly, to a method and system for enabling a client system to gain access to the remote database for workload balancing and resynchronization. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its

requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

5 In accordance with a preferred embodiment of the present invention, each DBMS member in the sysplex is assigned two network addresses. The first is a common network address shared by all DBMS members in the sysplex. The client utilizes the shared network address to access the sysplex via a standard router, which is capable of connecting the client to any active member located at the shared network address. The second network address assigned to each DBMS is member-specific and is attached to the member. Once the client has initiated contact with one member via the shared network address, that member provides a weighted list to the client that includes each active member's member-specific network address. The client can then distribute work evenly across the other active members. Moreover, if the connection between the client and a given member is interrupted during a commit process, the client can easily reconnect to the given member using that member's member-specific network address. Thus, work balancing and resynchronization is achieved without having to integrate a DNS with the sysplex system.

10
15
20 Figure 2 illustrates a sysplex environment 200 in accordance with a preferred embodiment of the present invention. As is shown, the sysplex 200 includes a plurality of networked computers 201-203 coupled to a coupling facility 220, which in turn is coupled to a pool 210 of storage devices 211-213 where the database resides. While each of the computers 201-203 can include a DBMS member 221-223, such is not a requirement. In a preferred embodiment of the present invention, a client 240 is coupled to a network router

250, which in turn is coupled to the sysplex network 200. It should be recognized that, although Figure 2 illustrates a single router 250/client 240 coupling, a plurality of routers and a plurality of clients can be coupled to the sysplex network 200.

According to a preferred embodiment, each DBMS member 221-223 is assigned two network addresses. The first network address is a shared network address (YYY), which is common to all members 221-223 of the sysplex 200. The shared network address YYY represents the entire database sharing location, and is used by the client 240 to access the database as a single image. The second network address assigned to each member 221-223 is a member-specific network address (XXA, XXB, XXC). No two members of the sysplex have the same member-specific network address. Both the shared and the member specific network addresses are linked to the corresponding DBMS member 221-223, and not to the computer 201-203 on which the member may reside. This becomes particularly relevant for resynchronization purposes.

The shared network address (YYY) is configured to access any DBMS member 221-223 of the sysplex 200. Therefore, so long as one DBMS member 221-223 is active, the network router 250 will connect the client 240 to one of the active members 221-223. In one preferred embodiment, the network router 250 is in direct communication with each DBMS member 221-223 and is aware of each member's status, i.e., active or inactive. Thus, the router 250 can route directly to one of the active members 221-223. In another embodiment, the network router 250 is coupled to a WLM 224, from which the same status information can be extracted.

A DNS 230 is coupled to the client 240, but is not integrated into the sysplex network 200. As before, the DNS 230 keeps a directory of database location domain names.

In the preferred embodiment of the present invention, the DNS 230 need only store the shared network address of each remote database location. Thus, when the client 240 wishes to access a remote database using the database location domain name, the DNS 230 resolves the domain name to one network address, i.e., the shared network address, and returns that address to the client 240.

Once the client 240 has established contact with one of the active DBMS members, e.g., 221, a database protocol takes over and returns a weighted list 260 of the active DBMS members 221-223. In a preferred embodiment, the weighted list 260 includes the respective workloads 261 of each member and each member's member-specific network address 262. The client 240 can then use the weighted list 260 to distribute new connections evenly across the active members 221-223 of the database sysplex 200, thereby balancing the workload.

Figure 3 illustrates a process for workload balancing in accordance with the preferred embodiment of the present invention. The process 300 begins by assigning a shared network address (YYY) to each DBMS member 221-223, via step 310. The shared network address (YYY) is the same for all members 221-223 of the sysplex 200. The client 240 asks the DNS 230 to resolve a database location domain name associated with a database residing in the sysplex 200, via step 320. The DNS 230 resolves the domain name to the shared network address (YYY), and returns it to the client 240, via step 330. In the alternative, the client 240 can maintain its own directory (not shown) that includes the shared network addresses of its databases.

In step 340, the client 240 asks the network router 250 to route to the shared network address (YYY) of the database. The router 250 determines which DBMS members 221-223, if any, are active, and connects the client 240 to any one of the active DBMS members 221-

223 in step 350. Once the client 240 is connected to one DBMS member, that DBMS member will return the weighted list 260 to the client 240 in step 360, who can then distribute work evenly across the active members using the member-specific network addresses 262 in step 370.

5 As explained above, the client 240 can direct work to a particular DBMS member, presumably the least loaded member, e.g. member 221, by routing the request or command to the member's member-specific network address (XXA). The member-specific network address also provides another advantage when communication between the client 240 and a member 221 has been interrupted during a two-phase commit procedure and the member 221 has been restarted on a different computer, e.g. computer 202 or 203.

10 As stated above, each DBMS member 221-223 "owns" a log dataset 251-253 that records every unit of work processed by the owning DBMS member. Only the DBMS member can access its log dataset, i.e. the information in the log dataset is not shared between the members. When a communication failure occurs during the two-phase commit process, the client 240 must "resynchronize" with the member of the sysplex, e.g., 221, that owns the log records 251 associated with the client's 240 unit of work. The resynchronization process allows the client 240 to determine the outcome (success or failure) of the unit of work at the DBMS member 221. In order to perform resynchronization, however, the client 240 must re-establish communications with the member 221 that performed the original unit of work.

20 According to the method and system of the present invention, the member-specific network address (XXA) is linked to its corresponding member, and not to the computer on which that member 221 may be running. The client 240 need only use the member-specific

network address (XXA) to reconnect to the member 221, regardless of where it may have been restarted. Thus, where it was once difficult to track down the network address of a particular member that had been restarted on a different computer, e.g., due to a computer failure, such a task is now simple under the method and system of the present invention.

5 Through aspects of the method and system of the present invention, workload balancing and commit resynchronization in a sysplex network can be performed without requiring a DNS to be tightly coupled to the sysplex. Problems associated with integrating a UNIX based application with a database application are avoided. By replacing the DNS with a network router, the latency associated with the DNS is eliminated and the time
10 required to establish an initial connection between the client and the sysplex is reduced, thereby improving performance. Moreover, no modifications to the client are necessary to implement the method and system of the present invention.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could
15 be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

For example, although the present invention has been described with reference to a sysplex environment, the invention could be applicable to other systems having multiple
20 servers which as those within a single system, e.g., one having multiple processors with shared memory, or a network of systems. Also, although this invention has been described with reference to a DBMS server such as a relational database management system product such as IBM DB2, the invention is applicable to any type of server, e.g., file servers, print

servers, file transfer programs, etc.

20100604 1005040